

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

Public Reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimates or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188,) Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 5/3/01	3. REPORT TYPE AND DATES COVERED Final report; end date 30 Apr 1997 01 Aug 95-
4. TITLE AND SUBTITLE Launch Support for the Skipper Satellite		5. FUNDING NUMBERS Contract DAAH04-95-1-0541
6. AUTHOR(S) Peter Erdman		8. PERFORMING ORGANIZATION REPORT NUMBER University of Pittsburgh; Pittsburgh, PA 15260
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U. S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211		10. SPONSORING / MONITORING AGENCY REPORT NUMBER 34575.1-EG-SDI
11. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.		
12 a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.		12 b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) Support was provided for a person to be at the Utah State Joint Operations Control Center (JOCC) during the flight of the Skipper satellite, and to develop software needed to display the spectrometers' data while the satellite is in orbit. A large amount of engineering and scientific data was to be collected during the flight of the Skipper satellite. These data needed to be reviewed in real time in order to evaluate the health of the satellite's subsystems and then to determine the next most reasonable step in the adaptive orbit strategy that was at the heart of the mission's design. In order to accomplish these ends, the scientific data needed to be disseminated to the members of the science team in absolute units for evaluation and interpretation. These tasks will required the development of specialized software for the conversion of the raw telemetry bit stream into calibrated data for graphic display in engineering units.		

14. SUBJECT TERMS		15. NUMBER OF PAGES 9	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OR REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION ON THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

20010516 112

Synopsis of Work:

The University of Pittsburgh, as a member of the science team, and as the designer and fabricator of the two scanning spectrometers aboard the Skipper satellite, received support for a person to be at the Utah State JOCC during the flight of Skipper, and to develop software needed to display the spectrometers' data while the satellite is in orbit.

A large amount of engineering and scientific data was to be collected during the flight of the Skipper satellite. These data needed to be reviewed in real time in order to evaluate the health of the satellite's subsystems and then to determine the next most reasonable step in the adaptive orbit strategy that was at the heart of the mission's design. In order to accomplish these ends, the scientific data needed to be disseminated to the members of the science team in absolute units for evaluation and interpretation. These tasks required the full time attention of a person expert with each instrument to insure both the proper functioning of the experiments and to apply the correct calibration factors to the raw telemetry bit stream so that selected data may be viewed in engineering units.

The understanding with the Russian science team members was to furnish them with the raw data and the instrument calibration data. However, without expert knowledge of the instrumentation itself, these data are nearly useless and perhaps even misleading. The proper application of the calibration data and interpretation of the results is critical for the retrieval of any science—and possibly even critical to convincing the Russian team members that they have received the complete and proper data sets.

Even the simplest of data is not easily interpreted without rather expert knowledge of the instrumentation that acquired it. In the case of the scanning spectrometers, there are a total of 25 different intensity calibrations that must be applied during the course of the flight due to the different operating modes. These intensity calibrations are in addition to the five different wavelength calibrations. All of this calibration data would be in a preliminary state at the time of the launch since it must be constructed from many dozens of laboratory data runs, some of which may not be so simply interpreted due to various problems which were not recognized at the time the data was taken.

The amount of data that must be handled is considerable—each pass, in fact, will have as much data as the entire Bowshock II flight. From this mass of data, selected regions must be extracted and viewed in detail, nearly in real time if the adaptive orbit strategy is to be useful. The software to accomplish this must be completed before the flight.

A widget-driven graphical interface was designed to accomplish these tasks using the high-level language "IDL" from Research Support Instruments. Over 1500 lines of code divided into 22 subroutines was written. This allowed the display and rapid manipulation of three data channels at a time from either of the Skipper spectrometers. The software project was a complete success. Simulations were run at the JOCC during payload integration that demonstrated the ability to rapidly extract the spectrometer data from the raw telemetry bit stream, review the spectrometer data, apply calibration factors, display the data in engineering units, subtract backgrounds, and determine the next course of action. The software was generic enough that it was extended past the requirements for the spectrometer alone, and had the additional ability to display the Utah State photometer data at the same time.

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

The following is included merely as a sample of the main widget-handler interface that was written for the project. The full code with all the subroutines is many times this length.

```

; BEGIN MAIN1           Skipper data, quick-look, widget interface  event handler

PRO MAIN1_Event, Event
  common sharewidgetnames, buttons, dependent1, fields, selmenuid, ptmenuid, zmmenuid,
  prmenuid      ;, FieldVal362 , field46
  common fileparameters, filename1, column, row, type
  common variables, data, numwindows, yranges, titles, posx, posy, selymax, calibstart,
  calibend, dataid, metadatax, metadatay, plothalf, semilog

  WIDGET_CONTROL,Event.Id,GET_UVALUE=Ev
  CASE Ev OF

    'PDMENU17': PDMENU17_Event, Event           ; Event for select file  menu
    'PDMENU21': PDMENU21_Event,Event             ; Event for lin/log display
    menu

    'PDMENU32': PDMENU32_Event, Event           ; Event for PDMENU32
    (plot menu)

    'PDMENU34': PDMENU34_Event, Event           ;Event for pdmenu34
    (zoom menu)

    'PDMENU40': PDMENU40_Event, Event           ;Event for pdmenu40  (print
    menu)

    ; 'BUTTON1': BEGIN                         ;former read file
    button, now part of select file
    ; readjocc          ;, column, row, type
    ; Print, 'Events for read file button'
    ; END

    'BUTTON2': BEGIN
      Print, 'Event for wshow'
      for i=1,numwindows do wshow,i           ;show the number of windows indicated by the
      variable determined in the file selection
      END

    'BUTTON3': BEGIN
      Print, 'Event for quit'
      ;print, 'active window just befofe quit= ', !d.window
      for i=1, numwindows+1 do wdelete, i       ;close all windows
      widget_control, /reset                   ;kill all widgets
      END

    'BUTTON4': BEGIN
      rezoom
      Print, 'Event for rezoom'
      END

    'BUTTON5': BEGIN
      cohersum
      Print, 'Event for co-sum'
      END

    'BUTTON6': BEGIN
      see=widget_message('ENTER ENCRYPTION KEYWORD', title='Calibration Privileges', /error)
      ; /error type message gives "stop" sign
      PRINT, 'Event for calib'
      END

    ; 'BUTTON7': BEGIN           ;plotter button replaced by printer pdmenu
    ; plothp
    ; PRINT, 'Event for plotting'
    ; END

    ENDCASE
  END
; END MAIN1

PRO PDMENU17_Event, Event

```

```

common sharewidgetnames, buttons, dependent1, fields, selmenuid, ptmenuid, zmmenuid,
prmenuid
common fileparameters, filename1, column, row, type
common variables, data, numwindows, yranges, titles, posx, posy, selymax, calibstart,
calibend, dataid, metadatay, metadatay, plothalf, semilog
;yranges=lonarr(10)  doesn't work to set data type to long integer!

CASE Event.Value OF
'select file.vuv spect.spectra': BEGIN
whichfil='*.vuv'
whichtit='Select Pdat Generated VUV  spectral data file'
column=6
row= 50000
type=1           ;type 1 for spectrometer data with fiducials, 0 without (i.e.
diagnostic data)           ; type 2 is "other" data (e.g. photometers)
numwindows=2
yranges=[65000, 5000, 10000, 1000, 1000, 10]           ;default plot maxima
(first element is a dummy to offset index and force type to long integer)
titles=['0', 'VUV Detector #1 Digital', 'VUV Detector #2 Digital', 'VUV Detector #1
Analog', 'VUV Detector #2 Analog', '0']
widget_control, ptmenuid(1), sensitive=1, set_value='digital data'           ;set names
of "plot" buttons
widget_control, ptmenuid(2), sensitive=1, set_value='analog data'
findfile, whichfil, whichtit
END

'select file.vuv spect.diagnost1': BEGIN
whichfil='*.vd1'
whichtit='Select Pdat Generated VUV diagnostic1 data file'
column=8
row= 10000
type=0           ;type 1 for data with fiducials, 0 without (i.e. diagnostic data)
           ; type 2 is "other" data (e.g. photometers)
numwindows=3
yranges=[5, 5, 5, 5, 5, 5, 5]
titles=['0', 'VUV +28', 'VUV +40', 'VUV +15', 'VUV -15', 'VUV +5', 'VUV PS Temp']
widget_control, ptmenuid(1), sensitive=1, set_value='vuv pwer sup a'           ;set names
of "plot" buttons
widget_control, ptmenuid(2), sensitive=1, set_value='vuv pwer sup b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.vuv spect.diagnost1'
END

'select file.vuv spect.diagnost2': BEGIN
whichfil='*.vd2'
whichtit='Select Pdat Generated VUV diagnostic2 data file'
column=8
row= 10000
type=0           ;type 1 for data with fiducials, 0 without (i.e. diagnostic data)
numwindows=3
yranges=[5, 5, 5, 5, 5, 5, 5]
titles=['0', 'VUV +28', 'vuv motor enab', 'VUV spec Temp', 'VUV HV1', 'VUV HV2', 'VUV
+28' ]
widget_control, ptmenuid(1), sensitive=1, set_value='vuv det diag a'           ;set names
of "plot" buttons
widget_control, ptmenuid(2), sensitive=1, set_value='vuv det diag b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.vuv spect.diagnost2'
END

'select file.uv spect.spectra': BEGIN
whichfil='*.uv'
whichtit='Select Pdat Generated UV spectral data file'
column=8
row= 60000
type=1           ;type 1 for spectrometer data with fiducials, 0 without (i.e.
diagnostic data)           ; type 2 is "other" data (e.g. photometers)
numwindows=3
yranges=[65000, 20000, 20000, 10000, 1000, 1000, 1000, 1]
titles=['0', 'UV Detector #1 Digital', 'UV Detector #2 Digital', 'UV Detector #3 Digital',
$ 'UV Detector #1 Analog', 'UV Detector #2 Analog', 'UV Detector #3 Analog', '0']
widget_control, ptmenuid(1), sensitive=1, set_value='digital data'

```

```

widget_control, ptmenuid(2), sensitive=1 , set_value='analog data'
findfile, whichfil, whichtit
END

'select file.uv spect.diagnost1': BEGIN
whichfil='*.ud1'
whichtit='Select Pdat Generated UV diagnostic1 data file'
column=8
row= 10000
type=0
numwindows=3
yranges=[5, 5, 5, 5, 5, 5]
titles=['0', 'UV +28', 'UV +40', 'UV +15', 'UV -15', 'UV +5', 'UV PS Temp']
,titles of diagnostic data1
widget_control, ptmenuid(1), sensitive=1 , set_value='uv power sup a' ;set names
of "plot" buttons
widget_control, ptmenuid(2), sensitive=1 , set_value='uv power sup b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.uv spect.diagnost1'
END

'select file.uv spect.diagnost2': BEGIN
whichfil='*.ud2'
whichtit='Select Pdat Generated UV diagnostic2 data file'
column=8
row= 10000
type=0
numwindows=3
yranges=[5, 5, 5, 5, 5, 5]
titles=['0', 'UV +28', 'UV motor enab', 'UV spec Temp', 'UV HV1', 'UV HV2', 'UV HV3' ]
,titles of diagnostic data2
widget_control, ptmenuid(1), sensitive=1 , set_value='uv det diag a' ;set names
of "plot" buttons
widget_control, ptmenuid(2), sensitive=1 , set_value='uv det diag b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.uv spect.diagnost2'
END

'select file.photometers.group1': BEGIN
whichfil='*.pg1'
whichtit='Select Pdat Generated Photometer Group #1 data file'
column=8
row= 30000
type=2 ;just means it isn't a type 0 or 1, each of which have special
treatments (presently only spectrometer data of those types)
; fiducials in type 1 for co-summations (spectrometer
detector data), and full scale of 5v for type 0 (spectrometer diagnostic data)
numwindows=3
yranges=[65000, 200, 200, 200, 200, 100, 500]
titles=['0', 'photometer #1, 230nm, 51.1nm bandwidth, low sensitivity, radial looking', '$
sensitivity, radial looking', '$
sensitivity', '$
sensitivity', '$
sensitivity'] ;photometer #2, 230nm, 51.1nm bandwidth, high
;photometer #3, 230nm, 51.1nm bandwidth, aft looking',
;photometer #4, 215nm, 6.4nm bandwidth, high
;photometer #5, 215nm, 6.4nm bandwidth, low
;photometer #6, 309nm, 11.3nm bandwidth, low
widget_control, ptmenuid(1), sensitive=1 , set_value='Group 1a' ;set names of
"plot" buttons
widget_control, ptmenuid(2), sensitive=1 , set_value='Group 1b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.photometers.group1'
END

'select file.photometers.group2': BEGIN
whichfil='*.pg2'
whichtit='Select Pdat Generated Photometer Group #2 data file'
column=8
row= 30000
type=2 ;just means it isn't a type 0 or 1 which each have special treatments
(presently only spectrometer data of those types)

```

```

; fiducials in type 1 for co-summations (spectrometer
detector data), and full scale of 5v for type 0 (spectrometer diagnostic data)
numwindows=3
yranges=[65000, 200, 200, 200, 300, 500, 3000]
titles=['0', 'photometer #7, 309nm, 11.3nm bandwidth, high sensitivity', $
        'photometer #8, 280nm, 4.7nm bandwidth, radial looking', $
        'photometer #9, 230nm, 51.1nm bandwidth, low
sensitivity', $
        'photometer #10, 230nm, 51.1nm bandwidth, high
sensitivity', $
        'photometer #11, 540nm, 4.6nm bandwidth, radial looking', $
        'photometer #12, 520nm, 29nm bandwidth']
widget_control, ptmenuid(1), sensitive=1, set_value='Group 2a' ;set names of
"plot" buttons
widget_control, ptmenuid(2), sensitive=1, set_value='Group 2b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.photometers.group2'
END

'select file.photometers.group3': BEGIN
whichfil='*.pg3'
whichtit='Select Pdat Generated Photometer Group #3 data file'
column=8
row= 30000
type=2 ;just means it isn't a type 0 or 1 which each have special treatments
(presently only spectrometer data of those types)
; fiducials in type 1 for co-summations (spectrometer
detector data), and full scale of 5v for type 0 (spectrometer diagnostic data)
numwindows=3
yranges=[65000, 200, 200, 200, 200, 200, 2000]
titles=['0', 'photometer #13, 391nm, 5.1nm bandwidth', $
        'photometer #14, 385nm, 5.2nm bandwidth', $
        'photometer #15. 309nm, 11.3nm bandwidth, radial looking', $
        'photometer #16, 778nm, 6.0nm bandwidth, low sensitivity', $
        'photometer #17, 778nm, 6.0nm bandwidth, high
sensitivity', $
        'photometer #18, 709nm, 5.9nm bandwidth, radial looking']
widget_control, ptmenuid(1), sensitive=1, set_value='Group 3a' ;set names of
"plot" buttons
widget_control, ptmenuid(2), sensitive=1, set_value='Group 3b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.photometers.group3'
END

'select file.photometers.group4': BEGIN
whichfil='*.pg4'
whichtit='Select Pdat Generated Photometer Group #4 data file'
column=4
row= 30000
type=2 ;just means it isn't a type 0 or 1 which each have special treatments
(presently only spectrometer data of those types)
; fiducials in type 1 for co-summations (spectrometer
detector data), and full scale of 5v for type 0 (spectrometer diagnostic data)
numwindows=1
yranges=[65000, 2000, 2000, 200, 100, 100, 50]
titles=['0', 'photometer #19, 610nm, 5.5nm bandwidth', $
        'photometer #20, 540nm, 4.6nm bandwidth', $
        'photometer #21', 'photometer #22', 'photometer #23', 'photometer #24']
;
widget_control, ptmenuid(1), sensitive=1, set_value='Group 4a' ;set names of
"plot" buttons
widget_control, ptmenuid(2), sensitive=1, set_value='Group 4b'
findfile, whichfil, whichtit
PRINT, 'Event for select file.photometers.group4'
END

ENDCASE
END
; END PDMENU17

; BEGIN PDMENU32
PRO PDMENU32_Event, Event

```

```

    common sharewidgetnames, buttons, dependent1, fields, selmenuid, ptmenuid, zmmenuid,
prmenuid
    common fileparameters, filename1, column, row, type
    common variables, data, numwindows, yranges, titles, posx, posy, selymax, calibstart,
calibend, dataid, metadatay, metadatay, plothalf, semilog

    CASE Event.Value OF
    'plot.data a': BEGIN
    plothalf=0
plotbutt
    PRINT, 'Event for plot.data a'
    END
    'plot.data b': BEGIN
plothalf=1
plotbutt
    END
    ENDCASE
END
;  END PDMENU32

;  BEGIN PDMENU34
PRO PDMENU34_Event, Event
    common sharewidgetnames, buttons, dependent1, fields, selmenuid, ptmenuid, zmmenuid,
prmenuid
    common fileparameters, filename1, column, row, type
    common variables, data, numwindows, yranges, titles, posx, posy, selymax, calibstart,
calibend, dataid, metadatay, metadatay, plothalf, semilog

    CASE Event.Value OF
    'zoom.window 1': BEGIN
zwin=1
datazoom, zwin
print, 'event for zoom.window1'
    end
    'zoom.window 2': BEGIN
zwin=2
datazoom, zwin
print, 'event for zoom.window2'
    end
    'zoom.window 3': BEGIN
zwin=3
datazoom, zwin
print, 'event for zoom.window3'
    end
    endcase
END
;end pdmenu34

;  BEGIN PDMENU40
PRO PDMENU40_Event, Event
    common sharewidgetnames, buttons, dependent1, fields, selmenuid, ptmenuid, zmmenuid
    common fileparameters, filename1, column, row, type
    common variables, data, numwindows, yranges, titles, posx, posy, selymax, calibstart,
calibend, dataid, metadatay, metadatay, plothalf, semilog

    CASE Event.Value OF
    'print.plotter': BEGIN
    plotp
    PRINT, 'Event for print.plotter'
    END
    'print.file': BEGIN
    fileout
    PRINT, 'Event for print.file'
    END
    ENDCASE
END
;  END PDMENU40

;  BEGIN DEP45
PRO DEP45_Event, Event
    common sharewidgetnames, buttons, dependent1, fields, selmenuid, ptmenuid, zmmenuid,
prmenuid
    common variables, data, numwindows, yranges, titles, posx, posy, selymax, calibstart,
calibend, dataid, metadatay, metadatay, plothalf, semilog
    WIDGET_CONTROL,Event.Id,GET_UVALUE=EV

```

```

CASE Ev OF
  'FIELD46': BEGIN
    Print, 'Event for plot1 max'
    END
  'FIELD47': BEGIN
    Print, 'Event for plot2 max'
    END
  'FIELD48': BEGIN
    Print, 'Event for plot3 max'
    END
  'BUTTON50': BEGIN
    print, 'Event for display'
    semilog=NOT(semilog) ;originally set semilog to 0 (for linear
display), so toggles between 0 and -1
    print, 'semilog value=', semilog
    if semilog eq (-1) then widget_control, fields(3), set_value='semilog display' else
    widget_control, fields(3), set_value='linear display'
    END
  'BUTTON49': BEGIN
    widget_control, fields(0), get_value=yrange1
    yranges(1+numwindows*plothalf)=yrange1
    widget_control, fields(1), get_value=yrange2
    yranges(2+numwindows*plothalf)=yrange2
    widget_control, fields(2), get_value=yrange3
    yranges(3+numwindows*plothalf)=yrange3
    WIDGET_CONTROL, DEPendent1, map =0 ;causes dependent base (plot yranges) to
disappear
    plotdat2 ;plot the data
    Print, 'Event for yranges done'
    END
  ENDCASE
END
; END DEP45

```

```

PRO JOCCWID, GROUP=Group
  common sharewidgetnames, buttons, dependent1, fields, selmenuid, ptmenuid, zmmenuid,
  prmenuid
  common variables, data, numwindows, yranges, titles, posx, posy, selymax, calibstart,
  calibend, dataid, metadatay, metadatay, plothalf, semilog

```

```
IF N_ELEMENTS(Group) EQ 0 THEN GROUP=0
```

```
junk = { CW_PDMENU_S, flags:0, name:'' } ;just defines the
structure?
```

```
MAIN1 = WIDGET_BASE(GROUP_LEADER=Group, $
  ROW=1, $
  MAP=1, $
  TITLE='Skipper data', $
  UVALUE='MAIN1', $
  XOFFSET=1010)
```

```
BASE13 = WIDGET_BASE(MAIN1, $
  COLUMN=1, $
  MAP=1, $
  TITLE='selectfile', $
  UVALUE='BASE13')
```

```
MenuDesc104 = [ $
  { CW_PDMENU_S, 3, 'select file' }, $; 0
  { CW_PDMENU_S, 1, 'vuv spect' }, $; 1
  { CW_PDMENU_S, 0, 'spectra' }, $; 2
  { CW_PDMENU_S, 0, 'diagnost1' }, $; 3
  { CW_PDMENU_S, 2, 'diagnost2' }, $; 4
  { CW_PDMENU_S, 1, 'uv spect' }, $; 5
  { CW_PDMENU_S, 0, 'spectra' }, $; 6
  { CW_PDMENU_S, 0, 'diagnost1' }, $; 7
  { CW_PDMENU_S, 2, 'diagnost2' }, $; 8
  { CW_PDMENU_S, 3, 'photometers' }, $; 9
  { CW_PDMENU_S, 0, 'group1' }, $; 10
  { CW_PDMENU_S, 0, 'group2' }, $; 11
  { CW_PDMENU_S, 0, 'group3' }, $; 12
  { CW_PDMENU_S, 2, 'group4' } ] ; 13
```

```

PDMENU17 = CW_PDMENU( BASE13, MenuDesc104, /RETURN_FULL_NAME, ids=selmenuid, $  

    UVALUE='PDMENU17')

; BUTTON1 = WIDGET_BUTTON( BASE13, $  

;     UVALUE='BUTTON1', $  

;     VALUE='read new file')

BUTTON2 = WIDGET_BUTTON( BASE13, $  

    UVALUE='BUTTON2', $  

    VALUE='wshow')

; MenuDesc21 = [ $  

;     { CW_PDMENU_S,      3, 'lin/log' }, $ ;      0  

;     { CW_PDMENU_S,      0, 'linear' }, $ ;      1  

;     { CW_PDMENU_S,      2, 'semilog' }] ;      2

;PDMENU21 = CW_PDMENU( BASE13, MenuDesc21, /RETURN_FULL_NAME, $  

are stored in the variable "prMENUID" ;pdmenu IDs  

;UVALUE='PDMENU21')

BUTTON3 = WIDGET_BUTTON( BASE13, $  

    UVALUE='BUTTON3', $  

    VALUE='quit')

BASE25 = WIDGET_BASE(MAIN1, $  

    COLUMN=1, $  

    MAP=1, $  

    TITLE='plotdata', $  

    UVALUE='BASE25')

MenuDesc112 = [ $  

    { CW_PDMENU_S,      3, 'plot' }, $ ;      0  

    { CW_PDMENU_S,      0, 'data a' }, $ ;      1  

    { CW_PDMENU_S,      2, 'data b' }] ;      2

PDMENU32 = CW_PDMENU( BASE25, MenuDesc112, /RETURN_FULL_NAME, ids=ptMENUID, $  

;pdmenu IDs are stored in the variable "ptMENUID"  

;UVALUE='PDMENU32')

MenuDesc114 = [ $  

    { CW_PDMENU_S,      3, 'zoom' }, $ ;      0  

    { CW_PDMENU_S,      0, 'window 1' }, $ ;      1  

    { CW_PDMENU_S,      0, 'window 2' }, $ ;      2  

    { CW_PDMENU_S,      2, 'window 3' }] ;      3

PDMENU34 = CW_PDMENU( BASE25, MenuDesc114, /RETURN_FULL_NAME, ids=zmMENUID, $  

IDs are stored in the variable "zmMENUID" ;pdmenu  

;UVALUE='PDMENU34')

BUTTON4 = WIDGET_BUTTON( BASE25, $  

    UVALUE='BUTTON4', $  

    VALUE='rezoom')

BUTTON5 = WIDGET_BUTTON( BASE25, $  

    UVALUE='BUTTON5', $  

    VALUE='co-sum')

BUTTON6 = WIDGET_BUTTON( BASE25, $  

    UVALUE='BUTTON6', $  

    VALUE='calib')

MenuDesc116 = [ $  

    { CW_PDMENU_S,      3, 'print' }, $ ;      0  

    { CW_PDMENU_S,      0, 'plotter' }, $ ;      1  

    { CW_PDMENU_S,      2, 'file' }] ;      2

PDMENU40 = CW_PDMENU( BASE25, MenuDesc116, /RETURN_FULL_NAME, ids=prMENUID, $  

;pdmenu IDs are stored in the variable "prMENUID"  

;UVALUE='PDMENU40')

; BUTTON7 = WIDGET_BUTTON( BASE25, $  

;     UVALUE='BUTTON7', $

```

```

;      VALUE='print')

WIDGET_CONTROL, MAIN1, /REALIZE

DEPendent1 = WIDGET_BASE(GROUP_LEADER=Group, $
    COLUMN=1, $
    MAP=0, $ ;set default for this dependent base to be invisible.  do it
here to prevent screen flicker which occurs if realized with map=0
    title='plot maxima', $
    UVALUE='DEP45')

FIELD46 = CW_FIELD( DEPendent1,VALUE=yrange1, $
    ROW=1, $
    INTEGER=1, $
    long=1, $
    TITLE='plot1 max', $
    UVALUE='FIELD46')

FIELD47 = CW_FIELD( DEPendent1,VALUE=yrange2, $
    ROW=1, $
    INTEGER=1, $
    long=1, $
    TITLE='plot2 max', $
    UVALUE='FIELD47')

FIELD48 = CW_FIELD( DEPendent1, value=yrange3, $
    ROW=1, $
    INTEGER=1, $
    long=1, $
    TITLE='plot3 max', $
    UVALUE='FIELD48')

BUTTON50 = WIDGET_BUTTON( DEPendent1, $
    UVALUE='BUTTON50', $
    VALUE='linear display')

BUTTON49 = WIDGET_BUTTON( DEPendent1, $
    UVALUE='BUTTON49', $
    VALUE='done')

numwindows=1 ;just give a default value so the variable is defined if try
to quit before selecting a file ;give the default value for linear plots
semilog=0

WIDGET_CONTROL, DEPendent1, /REALIZE ; which occurs if it is done in a single
statement --better to have map=0 above in creation statement

buttons=[0, 0, button2, button3, button4, button5, button6] ;button1 , button7 make
common statements simpler by having button names in an array
fields=[ field46, field47, field48, button50]
buttondi ;call button disable procedure to disable all but file
find and quit buttons

XMANAGER, 'DEP45', DEPendent1, /JUST_REG
XMANAGER, 'MAIN1', MAIN1

END

```